



# Aula 16 - Armazenamento em tempo real com o Firebase



## Módulo de Internet das Coisas

- Prof.



UNIVERSIDADE  
ESTADUAL DO CEARÁ



MINISTÉRIO DA  
CIÊNCIA, TECNOLOGIA  
E INOVAÇÃO





# IA

## Objetivos

- Introduzir o Firebase Realtime Database (RTDB)
- Criar um projeto de um RTDB
- Usar o ESP32 para armazenar e ler dados do Firebase RTDB



# IA

## Introdução ao Firebase [1]

- Produto Google Cloud: Platform as Service - PaS
- Real Time Database [2]
  - NoSQL
  - armazenamento em nuvem
  - notificação em tempo real com usuários/clientes
  - suporte offline
- Cloud Storage
  - armazenamento massivo de dados



Build better apps



Auth



Hosting



Cloud Functions



ML Kit



Cloud Firestore



Realtime Database



Cloud Storage



Improve app quality



Crashlytics



Performance Monitoring



Test Lab



Grow your app



Analytics



Remote Config



Predictions



A/B Testing



Cloud Messaging



Dynamic Links



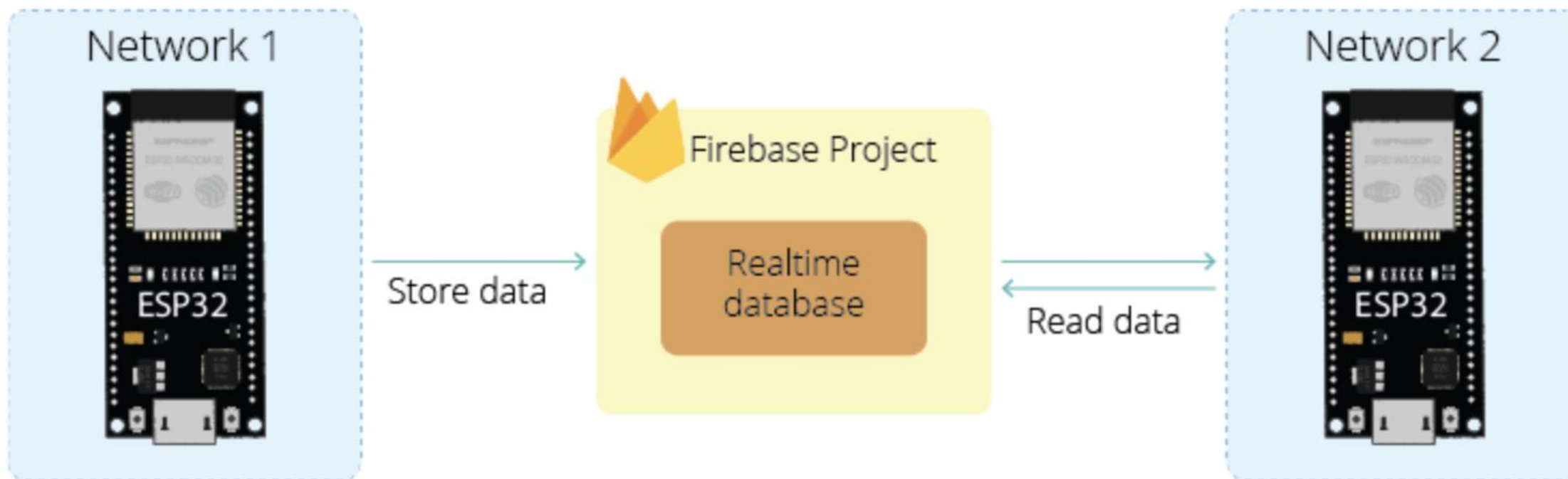
In-app Messaging



# IA

## Demo-Lab - Armazenar dados no Firebase RTDB com ESP32 [3]

1. Criar um projeto Firebase com um realtime database (RTDB)
2. Fazer o ESP32 armazenar dados no RTDB
3. Fazer o ESP32 ler dados do RTDB







# IA

## Demo-Lab - Armazenar dados no Firebase RTDB com ESP32 [3]

### 1 - Criar um projeto Firebase com um realtime database (RTDB)

1. Acesse [<https://firebase.google.com/?hl=pt-br>](https://firebase.google.com/?hl=pt-br) e entre com uma conta da Google
2. Clique em **Get Started** e depois **Add project** para criar um novo projeto
3. Dê um nome para set projeto, por exemplo *ESP32 Firebase Demo*
4. Desabilite a opção **Enable Google Analytics** para este projeto já que não é necessário e clique em **Create project**

× Create a project (Step 2 of 2)

### Google Analytics for your Firebase project

Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, Predictions, and Cloud Functions.

Google Analytics enables:

- × A/B testing ⓘ
- × User-segmentation & targeting across Firebase products ⓘ
- × Predicting user behavior ⓘ
- × Crash-free users ⓘ
- × Event-based Cloud Functions triggers ⓘ
- × Free unlimited reporting ⓘ

☐ Enable Google Analytics for this project  
Recommended

[Previous](#) [Create project](#)

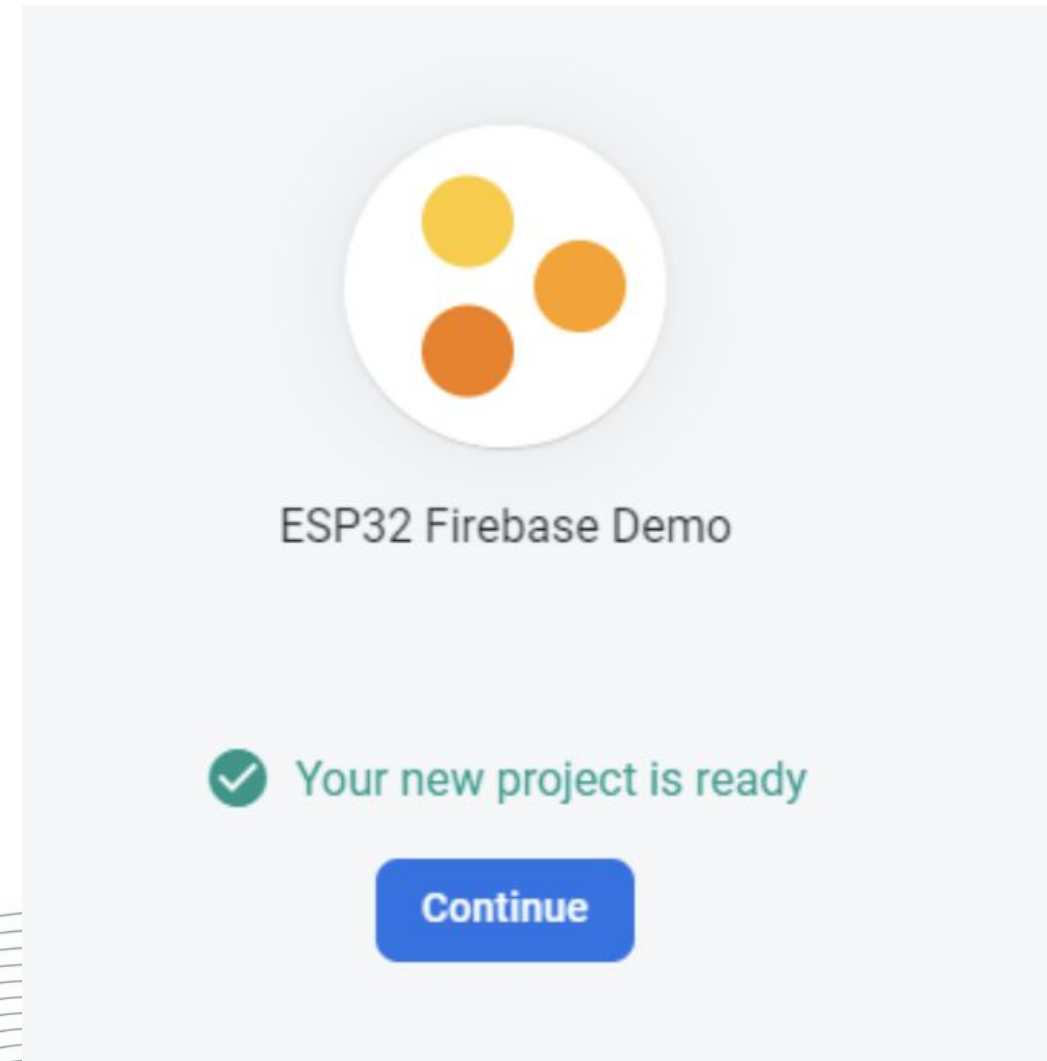


# IA

## Demo-Lab - Armazenar dados no Firebase RTDB com ESP32 [3]

1 - Criar um projeto Firebase com um realtime database (RTDB)

5 - Vai demorar alguns segundos para configurar o seu projeto.



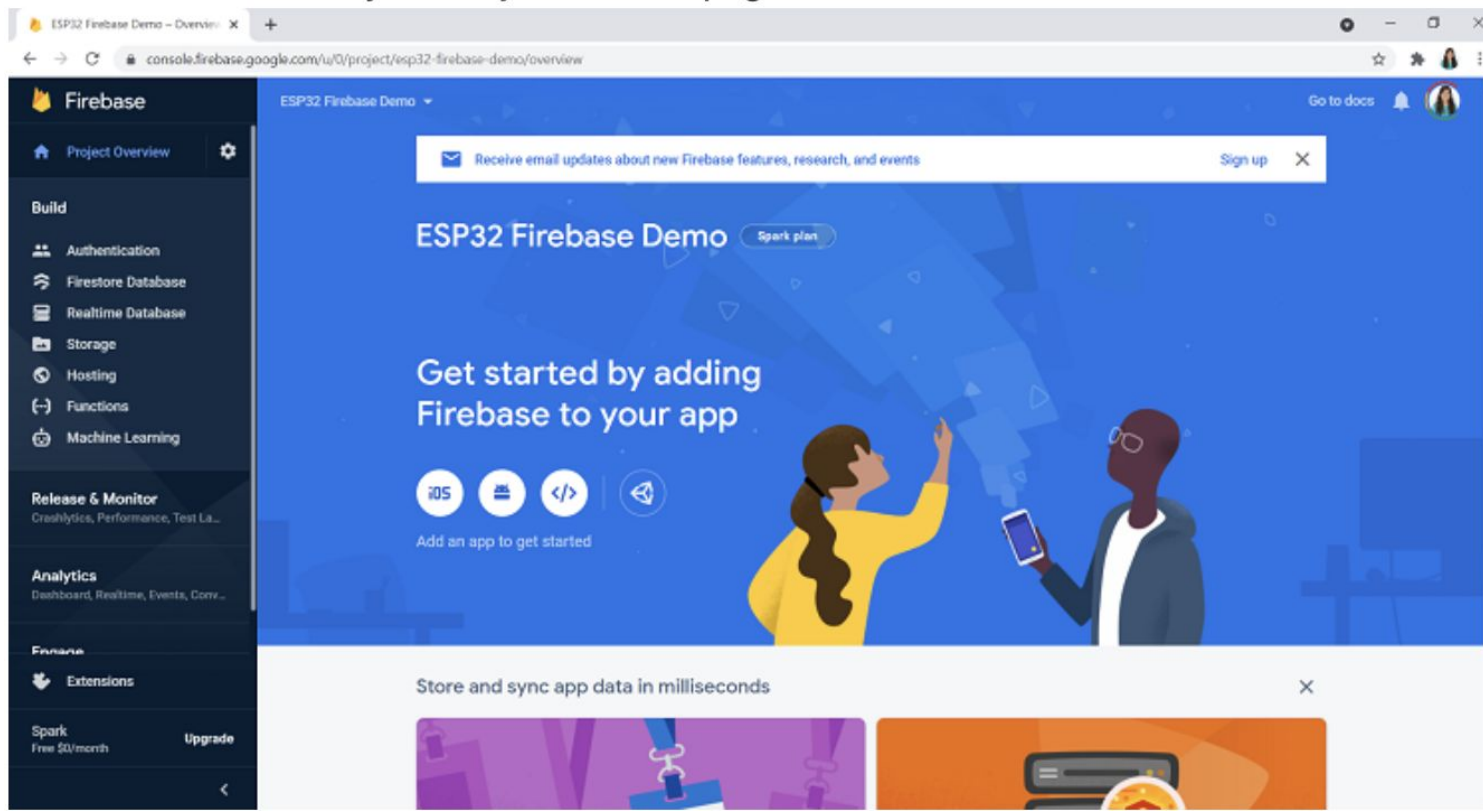


# IA

## Demo-Lab - Armazenar dados no Firebase RTDB com ESP32 [3]

### 1 - Criar um projeto Firebase com um realtime database (RTDB)

Você será redirecionado para a página console do seu Projeto





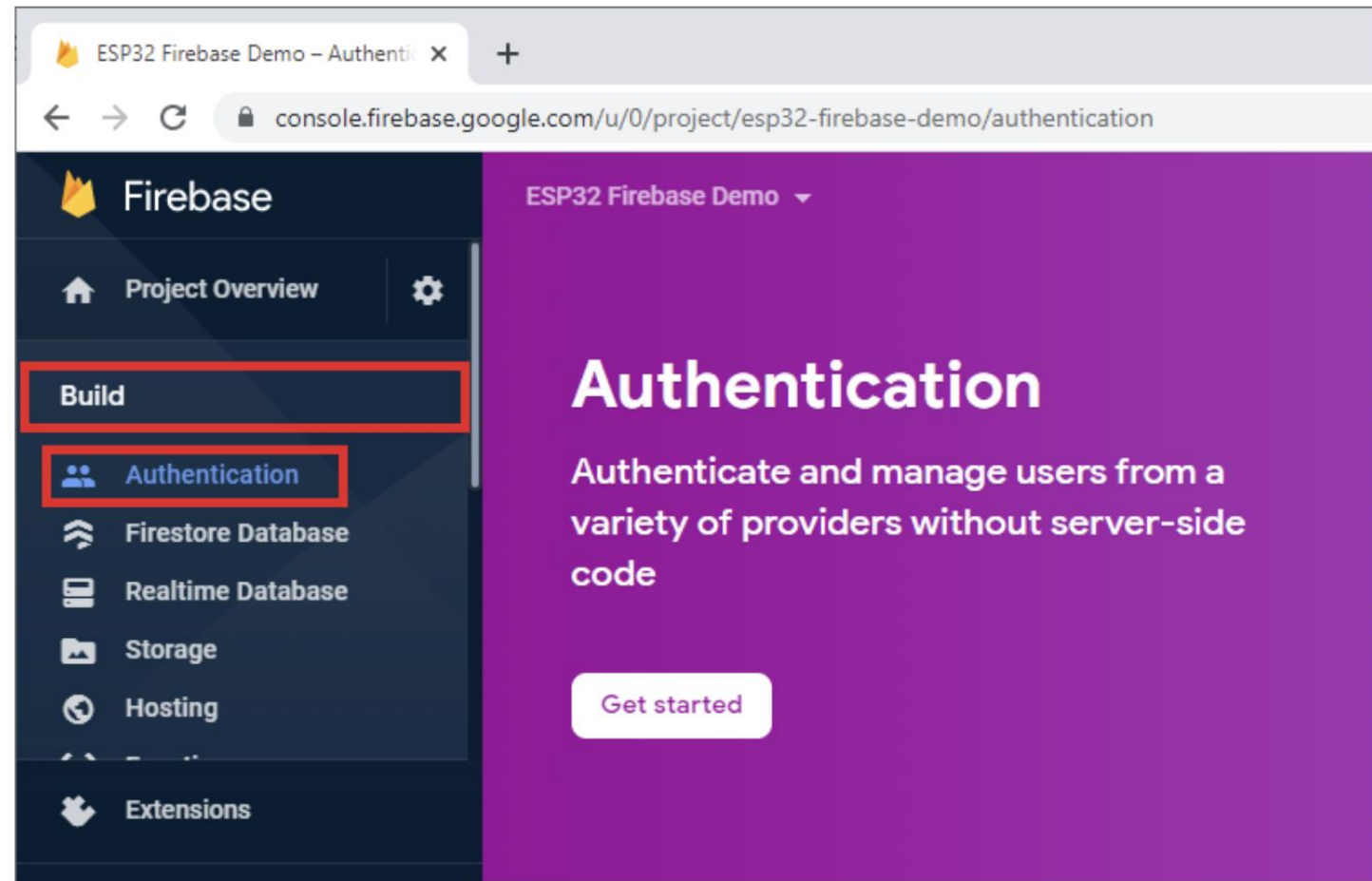
# IA

## Demo-Lab - Armazenar dados no Firebase RTDB com ESP32 [3]

### 2 - Configurar métodos de Autenticação

- permite que a aplicação na nuvem salve dados de um usuário (ex. ESP32) na nuvem

1 - No menu à esquerda, clique em **Authentication** e depois **Get Started**

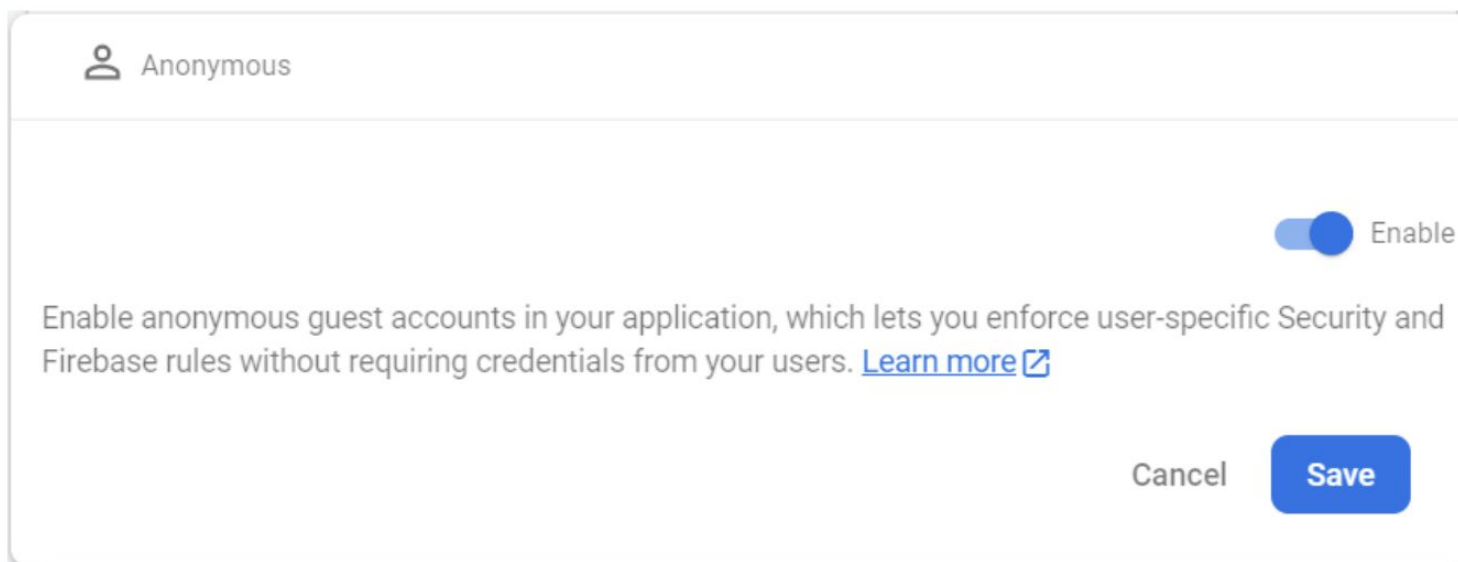






### 2 - Configurar métodos de Autenticação

1. Existem vários métodos de autenticação como email e senha, Google Account, Facebook e outros.
2. Com o propósito de teste, vamos selecionar o ***Anonymous***, que requer autenticação sem que o usuário se log primeiro, ou seja, criando contas anônimas temporárias. Habilite essa opção e clique em ***Save***.



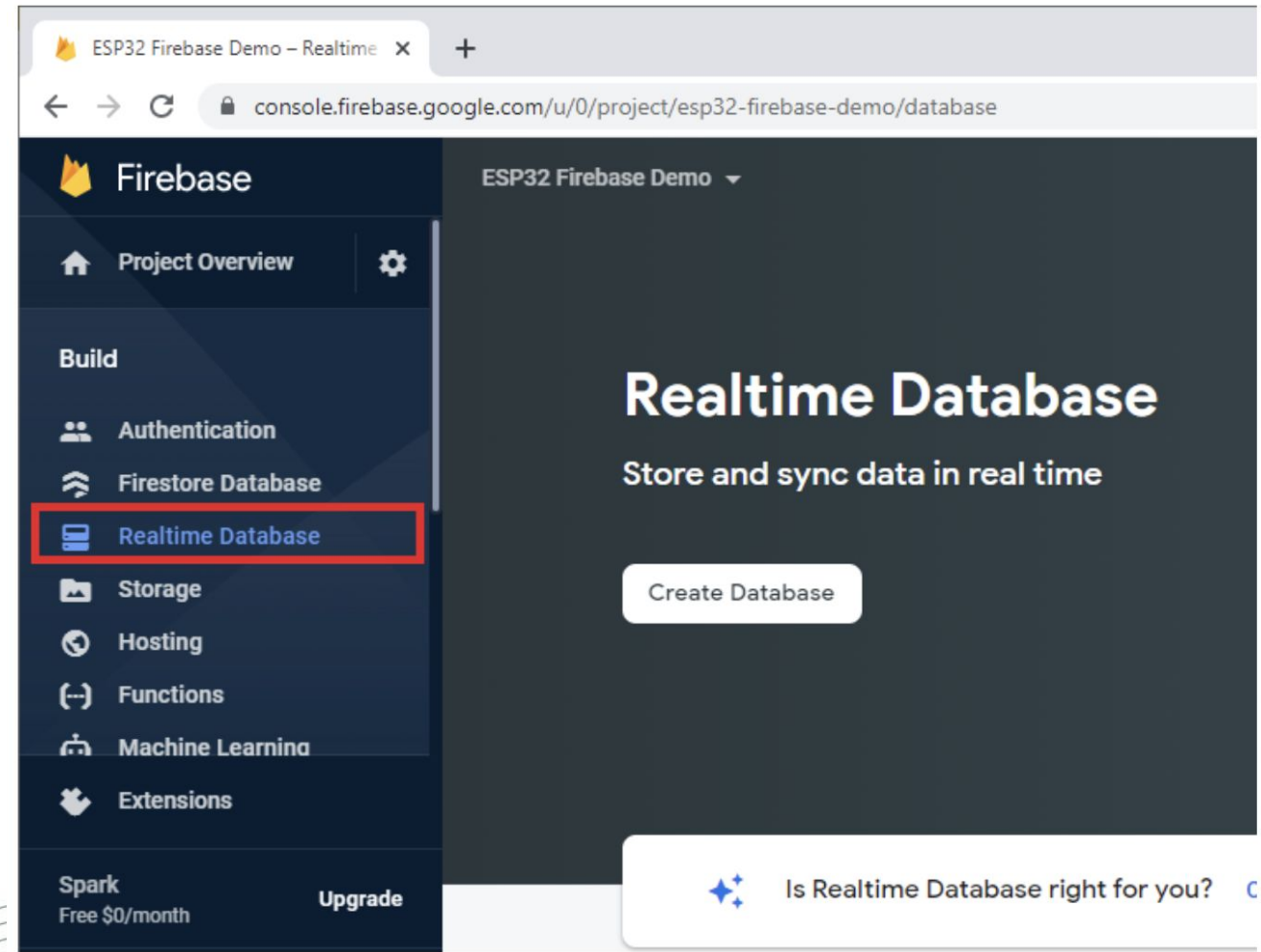


# IA

## Demo-Lab - Armazenar dados no Firebase RTDB com ESP32 [3]

### 3 - Criar um Reatime Database

1 - No menu à esquerda clique em *Reatime Database* e, em seguida, *Create Database*





# IA

## Demo-Lab - Armazenar dados no Firebase RTDB com ESP32 [3]

### 3 - Criar um Realtime Database

2. Selecione a localização do seu banco de dados o mais próximo possível de onde você está.

Your location setting is where your Realtime Database data will be stored.

Realtime Database location

Belgium (europe-west1)

Cancel

Next



# IA

## Demo-Lab - Armazenar dados no Firebase RTDB com ESP32 [3]

### 3 - Criar um Reatime Database

3. Configure as regras de segurança de seu banco de dados. Com propósito de teste selecione ***Start in test mode***.

### Set up database

1 Database options

2 Security rules

Once you have defined your data structure you will have to write rules to secure your data.  
[Learn more](#)

☐ Start in locked mode  
Your data is private by default. Client read/write access will only be granted as specified by your security rules.

☒ Start in test mode  
Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access.

```
{  
  "rules": {  
    ".read": "now < 1630537200000", // 2021-9-2  
    ".write": "now < 1630537200000", // 2021-9-2  
  }  
}
```

! The default security rules for test mode allow anyone with your database reference to view, edit and delete all data in your database for the next 30 days

Cancel

Enable



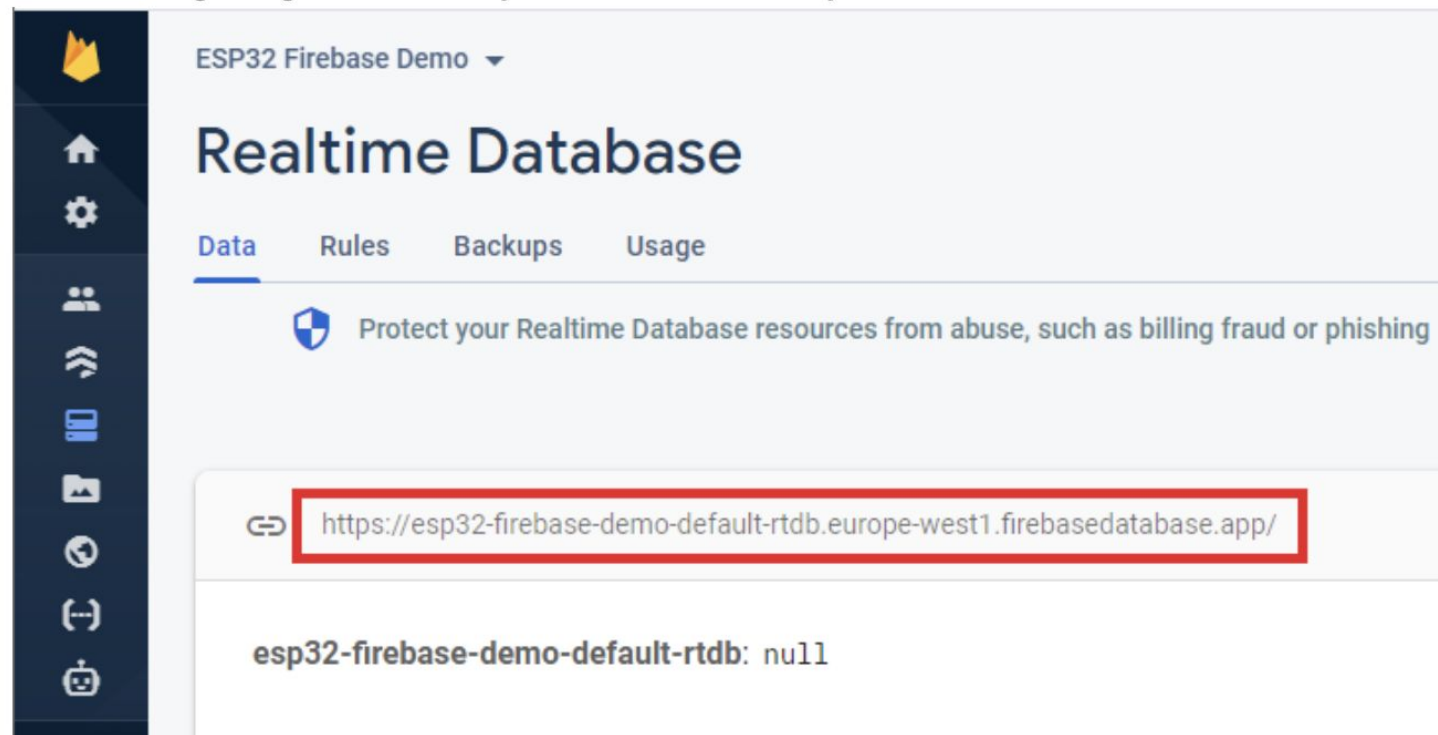


# IA

## Demo-Lab - Armazenar dados no Firebase RTDB com ESP32 [3]

### 3 - Criar um Reatime Database

4. Seu banco de dados foi criado. Você deve copiar e salvar a URL do banco de dados para que seja usado no código do ESP32.



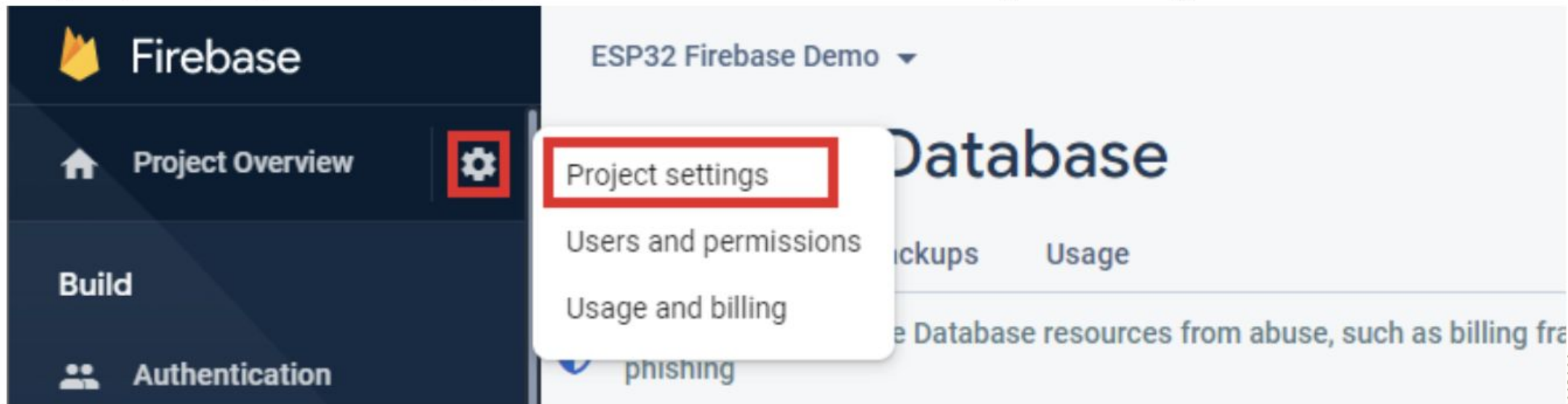


# IA

## Demo-Lab - Armazenar dados no Firebase RTDB com ESP32 [3]

### 4 - Obtenha a API Key do projeto

1. Para obter a chave do projeto, acesse o menu à esquerda, em **Project Settings**





# IA

## Demo-Lab - Armazenar dados no Firebase RTDB com ESP32 [3]

### 4 - Obtenha a API Key do projeto

2. Copie a API Key em um local seguro para utilizá-la posteriormente.

The screenshot shows the 'Project settings' page for a project named 'ESP32 Firebase Demo'. The 'General' tab is selected. The 'Web API Key' field is highlighted with a red box, showing the key: 'API key: AIzaSyB...'. The left sidebar shows the 'Project Overview' and various services like Authentication, Firestore Database, Realtime Database, Storage, Hosting, Functions, and Machine Learning.



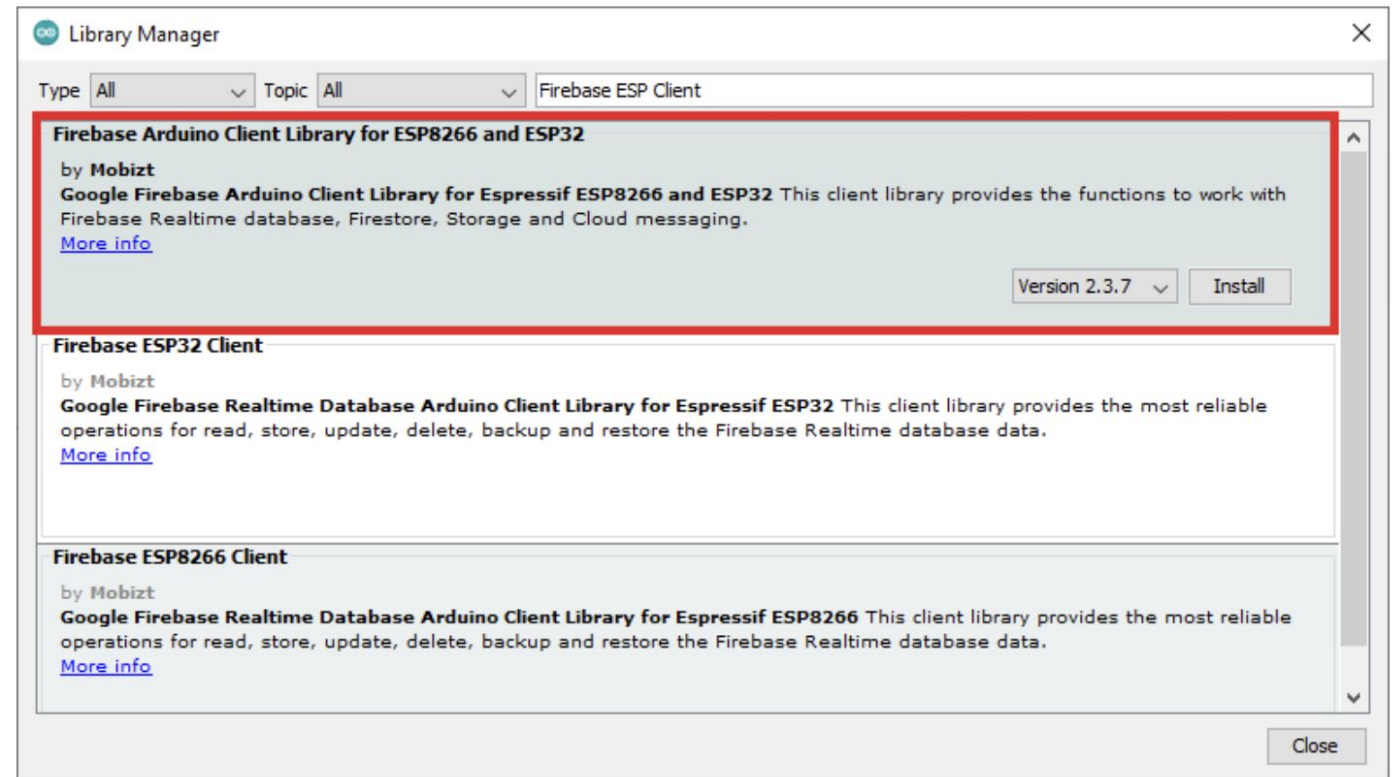
# IA

## Demo-Lab - Armazenar dados no Firebase RTDB com ESP32 [3]

5. Programar o ESP32 para acessar o Firebase

### Instalar a biblioteca Firebase-ESP32-Client

1. Na Arduino IDE, acesse *Sketch> Include library> Manage Libraries*
2. Procure Firebase ESP32 Client e instale a *Fireabase Arduino Client Library for ESP8266 e ESP32* by Mobitz



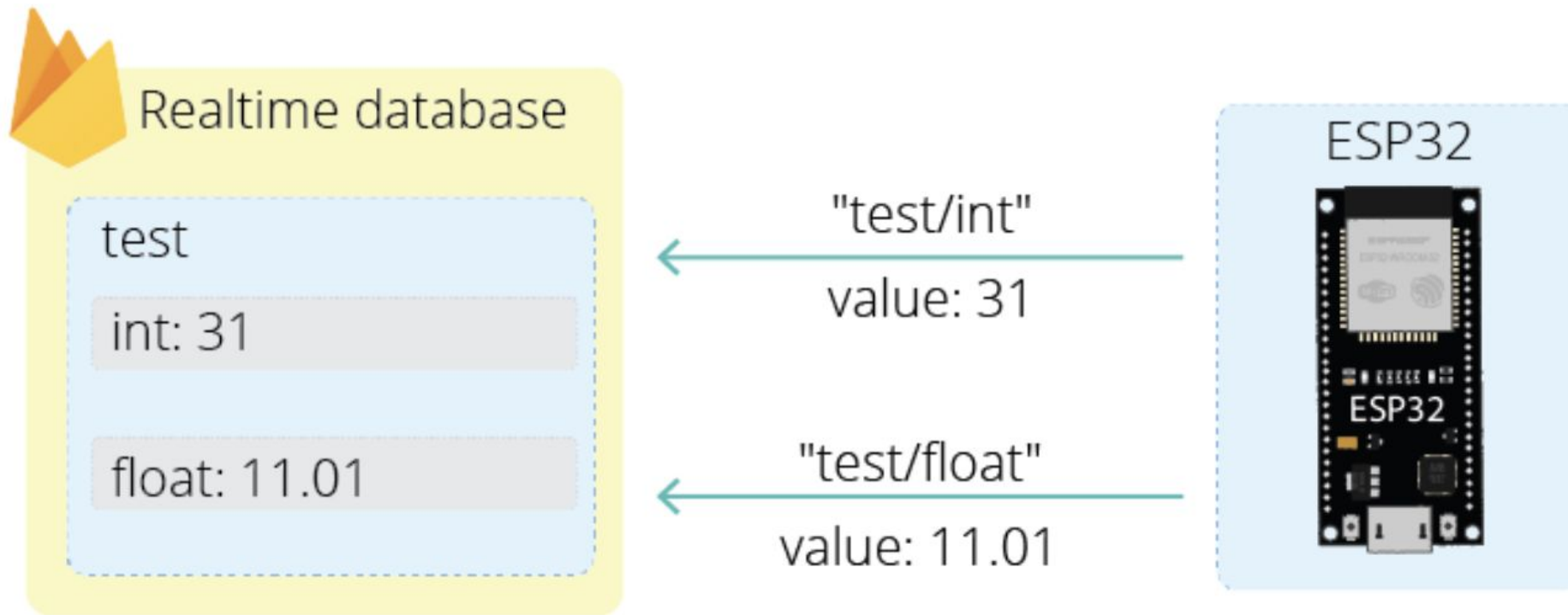




# IA

## Demo-Lab - Armazenar dados no Firebase RTDB com ESP32 [3]

### 5. Programar o ESP32 para armazenar dados no Firebase Baixar código exemplo do classroom





### 5. Programar o ESP32 para armazenar dados no Firebase

#### Bibliotecas

```
#include <Arduino.h>
#if defined(ESP32)
#include <WiFi.h>
#elif defined(ESP8266)
#include <ESP8266WiFi.h>
#endif
#include <Firebase_ESP_Client.h>
```

```
//Provide the token generation process info.
```

```
#include "addons/TokenHelper.h"
```

```
//Provide the RTDB payload printing info and other helper functions.
```

```
#include "addons/RTDBHelper.h"
```



# IA

## Demo-Lab - Armazenar dados no Firebase RTDB com ESP32 [3]

### 5. Programar o ESP32 para armazenar dados no Firebase

Modificar o código em:

```
#define WIFI_SSID "REPLACE_WITH_YOUR_SSID"  
#define WIFI_PASSWORD "REPLACE_WITH_YOUR_PASSWORD"
```

```
#define API_KEY "REPLACE_WITH_YOUR_FIREBASE_PROJECT_API_KEY"
```

```
#define DATABASE_URL "REPLACE_WITH_YOUR_FIREBASE_DATABASE_URL"
```





### 5. Programar o ESP32 para armazenar dados no Firebase setup(): conectar a Wifi

```
Serial.begin(115200);  
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);  
Serial.print("Connecting to Wi-Fi");  
while (WiFi.status() != WL_CONNECTED){  
    Serial.print(".");  
    delay(300);  
}  
Serial.println();  
Serial.print("Connected with IP: ");  
Serial.println(WiFi.localIP());  
Serial.println();
```





# IA

## Demo-Lab - Armazenar dados no Firebase RTDB com ESP32 [3]

### 5. Programar o ESP32 para armazenar dados no Firebase setup(): API Key e Database URL

```
/* Assign the api key (required) */  
config.api_key = API_KEY;  
  
/* Assign the RTDB URL (required) */  
config.database_url = DATABASE_URL;
```



### 5. Programar o ESP32 para armazenar dados no Firebase setup(): autenticação

```
/* Sign up */  
if (Firebase.signUp(&config, &auth, "", "")){  
  Serial.println("ok");  
  signupOK = true;  
}  
else{  
  Serial.printf("%s\n", config.signer.signupError.message.c_str());  
}
```



# IA

## Demo-Lab - Armazenar dados no Firebase RTDB com ESP32 [3]

### 5. Programar o ESP32 para armazenar dados no Firebase

loop(): envia dados periodicamente ao banco de dados quando autenticado

```
if (Firebase.ready() && signupOK && (millis() -  
sendDataPrevMillis > 15000 || sendDataPrevMillis == 0)) {
```



### 5. Programar o ESP32 para armazenar dados no Firebase

loop(): enviamos um inteiro para o caminho especificado e imprime sucesso ou falha.

```
if (Firebase.RTDB.setInt(&fbdo, "test/int", count)) {  
  Serial.println("PASSED");  
  Serial.println("PATH: " + fbdo.dataPath());  
  Serial.println("TYPE: " + fbdo.dataType());  
}  
else {  
  Serial.println("FAILED");  
  Serial.println("REASON: " + fbdo.errorReason());  
}
```





### 5. Programar o ESP32 para armazenar dados no Firebase

**loop():** enviamos um float para o caminho especificado e imprime sucesso ou falha.

```
// Write an Float number on the database path test/float
if (Firebase.RTDB.setFloat(&fbdo, "test/float", 0.01 + random(0, 100))) {
  Serial.println("PASSED");
  Serial.println("PATH: " + fbdo.dataPath());
  Serial.println("TYPE: " + fbdo.dataType());
}
else {
  Serial.println("FAILED");
  Serial.println("REASON: " + fbdo.errorReason());
}
```



# IA

## Demo-Lab - Armazenar dados no Firebase RTDB com ESP32 [3]

### 5. Programar o ESP32 para armazenar dados no Firebase

#### Demonstração

Faça o Upload do código para o ESP32 e abra o Serial Monitor com 115200 de baud rate

```
COM16  
| Send  
PASSED  
PATH: test/int  
TYPE: int  
PASSED  
PATH: test/float  
TYPE: float  
Autoscroll Show timestamp Newline 115200 baud Clear output
```



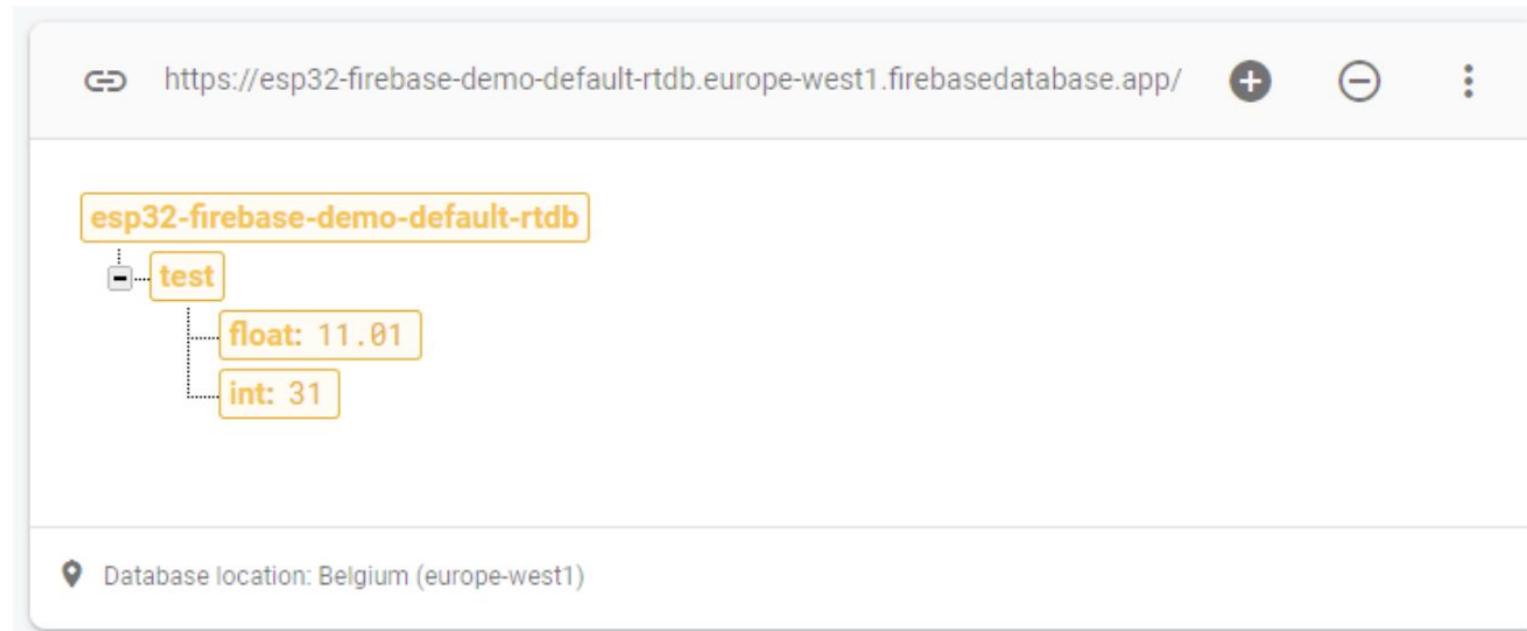
# IA

## Demo-Lab - Armazenar dados no Firebase RTDB com ESP32 [3]

### 5. Programar o ESP32 para armazenar dados no Firebase

#### Demonstração

No seus projetos do Firebase, você poderá ver valores em node paths sendo atualizados a cada 15 segundos. O database pisca quando valores são atualizados.

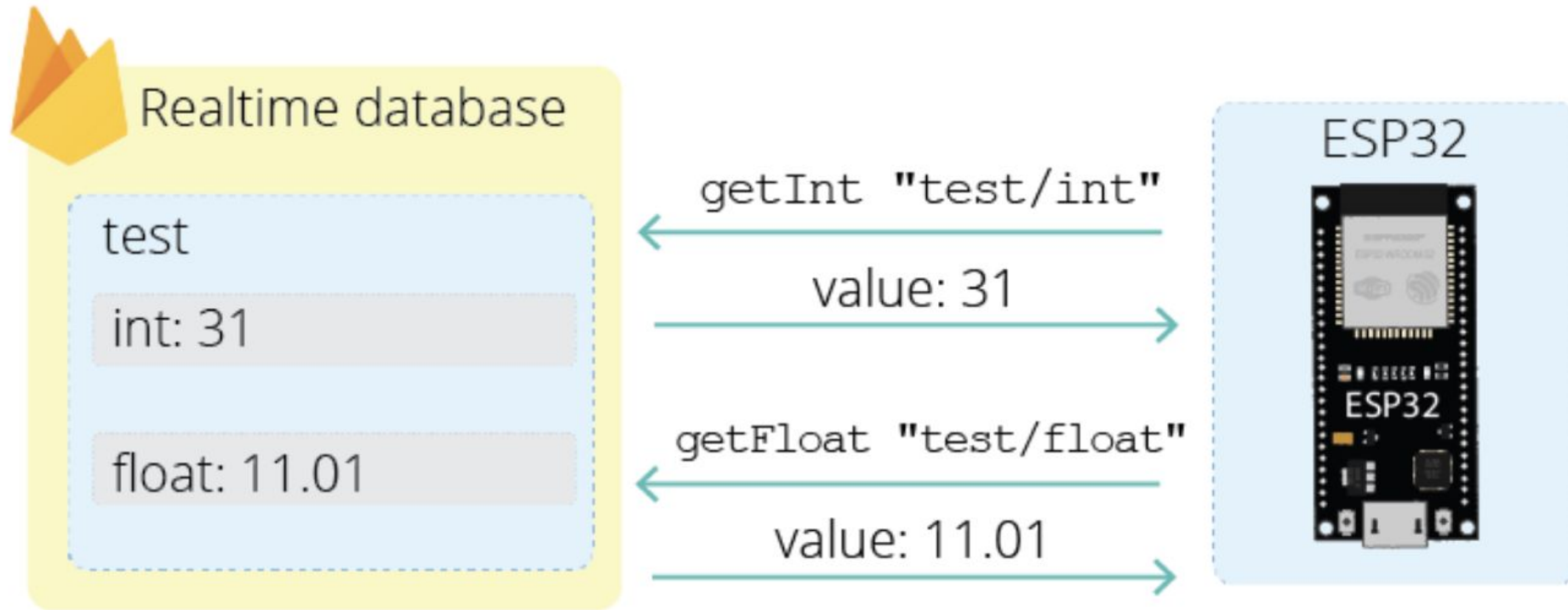




# IA

## Demo-Lab - Armazenar dados no Firebase RTDB com ESP32 [3]

### 6. Programar o ESP32 para ler dados no Firebase Baixar o código exemplo do classroom







# IA

## Demo-Lab - Armazenar dados no Firebase RTDB com ESP32 [3]

### 6. Programar o ESP32 para ler dados no Firebase

Modificar o código em:

```
#define WIFI_SSID "REPLACE_WITH_YOUR_SSID"  
#define WIFI_PASSWORD "REPLACE_WITH_YOUR_PASSWORD"
```

```
#define API_KEY "REPLACE_WITH_YOUR_FIREBASE_PROJECT_API_KEY"
```

```
#define DATABASE_URL "REPLACE_WITH_YOUR_FIREBASE_DATABASE_URL"
```



### 6. Programar o ESP32 para ler dados no Firebase ler um inteiro

```
if (Firebase.RTDB.getInt(&fbdo, "/test/int")) {  
  if (fbdo.dataType() == "int") {  
    intValue = fbdo.intData();  
    Serial.println(intValue);  
  }  
}  
else {  
  Serial.println(fbdo.errorReason());  
}
```



## 6. Programar o ESP32 para ler dados no Firebase

### ler um float

```
if (Firebase.RTDB.getFloat(&fbdo, "/test/float")) {  
  if (fbdo.dataType() == "float") {  
    floatValue = fbdo.floatData();  
    Serial.println(floatValue);  
  }  
}  
else {  
  Serial.println(fbdo.errorReason());  
}
```

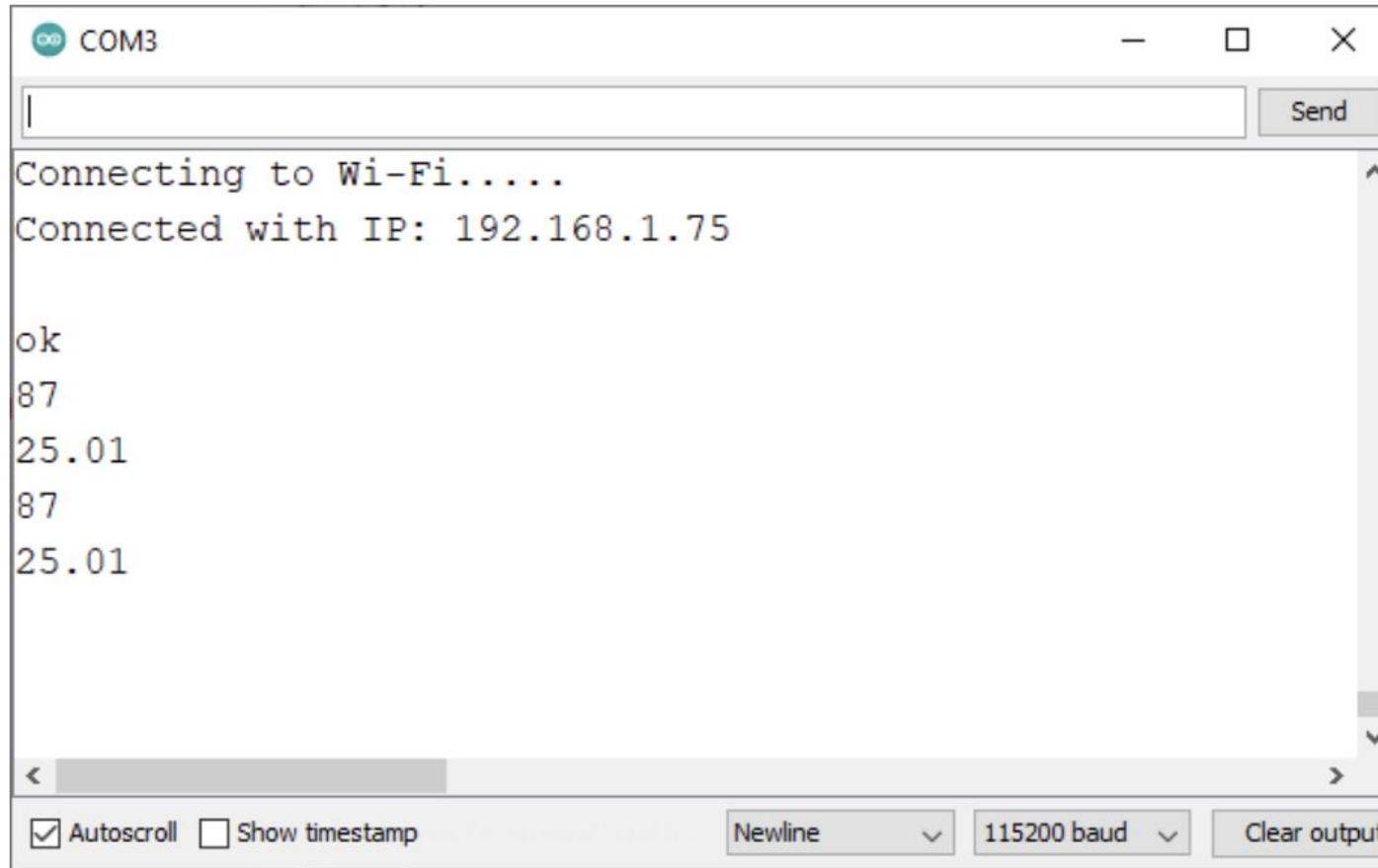


# IA

## Demo-Lab - Armazenar dados no Firebase RTDB com ESP32 [3]

### 6. Programar o ESP32 para ler dados no Firebase

#### Demonstração



```
COM3
|
|
Connecting to Wi-Fi.....
Connected with IP: 192.168.1.75

ok
87
25.01
87
25.01

[Autoscroll checked] [Show timestamp unchecked] [Newline] [115200 baud] [Clear output]
```





# IA

## Referência bibliográfica

[1]

<https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0>

[2]

<https://firebase.google.com/products/realtime-database?hl=pt-br>

[3]

<https://randomnerdtutorials.com/esp32-firebase-realtime-database/>

# Dúvidas?

Módulo de Internet das Coisas



UNIVERSIDADE  
ESTADUAL DO CEARÁ



Instituto Iracema  
PESQUISA E INOVAÇÃO



Softex

MINISTÉRIO DA  
CIÊNCIA, TECNOLOGIA  
E INOVAÇÃO

